**University of Virginia**

**Charles L. Brown Department of Electrical and Computer Engineering**

**Design Review 2: An Innovative High Speed Cache Design**

Jacob Breiholz, Leiqing Cai, Ashley Morse, Qing Qin

1. **INTRODUCTION**

In this design review, we present the progress for the design and implementation of the high speed cache which we purposed earlier on.

1. **TIME LINE**

Table 1 Previously Purposed Schedule with Updates

|  |  |  |
| --- | --- | --- |
| **Week** | **Milestone** | **Update** |
| Oct 17 - Oct 23 | Block Integration and Optimization | Block Integration Completed |
| Oct 24 - Oct 30 | Modeling and Architecture Optimization | Modeling Completed |
| Oct 31 - Nov 6 | Entire SRAM Integration | SRAM netlist Completed |
| Nov 7 - Nov 13 | Low-Voltage Operation | Optimized Decoder. |
| Nov 14 - Nov 20 | Leakage Reduction Mode |  |
| Nov 21 - Nov 27 | Layout | Finished SRAM array layout and sense amplifier layout. |
| Nov 28 - Dec 2 | Final Report and Presentations |  |

Table 2 Purposed Task Assignment with Updates

|  |  |  |
| --- | --- | --- |
| **Name** | **Project Task** | **Update** |
| Jacob Breiholz | Sense Amplifier Design | COMPLETED |
| Layout | IN PROGRESS |
| Leiqing Cai | Block Design and Optimization | COMPLETED |
| Clock and Timing | COMPLETED - Added clock buffers and registers for primary inputs. |
| Ashley Morse | Decoder Design | COMPLETED – Picked 3to8 static NOR pre dec. for Row Decoder, and 4to16 static NOR block decoder |
| Process Corner Simulation | IN PROGRESS |
| Qing Qin | Architecture | COMPLETED |
| Overall Simulation | COMPLETED |

Table 3 Newly Purposed Schedule

|  |  |  |
| --- | --- | --- |
| **Dates** | **Task** | **Assignee** |
| Nov 8 - Nov 14 | Finish Decoder Optimization  Finish layout for SRAM  Finish simulating on all process corners  Start Implementing Special Features  Turn in Project Proposal | Ashley  Jacob  Ashley, Leiqing, Qing  Leiqing, Qing  All Members |
| Nov 15 – Nov 21 | Finish Layouts for peripherals and Model  Start optimizing for overall system for speed  Adjust peripherals to improve system  Find the optimal banking for the system | Jacob  All Members  Qing  Qing |
| Nov 22 - Nov 28 | Finish Special Feature  Simulate all global process corners  Simulate at different voltages  Simulate at different temperatures | Leiqing  Ashley  Jacob  Qing |
| Nov 29 - Dec 2 | Make last minute changes and ensure functionality  Update Wiki Completely  Turn in Final Design Report  Give Design Presentation | All Members  Ashley, Jacob  All Members  All Members |

1. **PROGRESS BY BLOCKS**
2. **SRAM Architecture**

Figure 1 demonstrates the architecture of the SRAM design. A list of signals and pins with annotation is shown in Table 4. All schematics have been done using netlist.

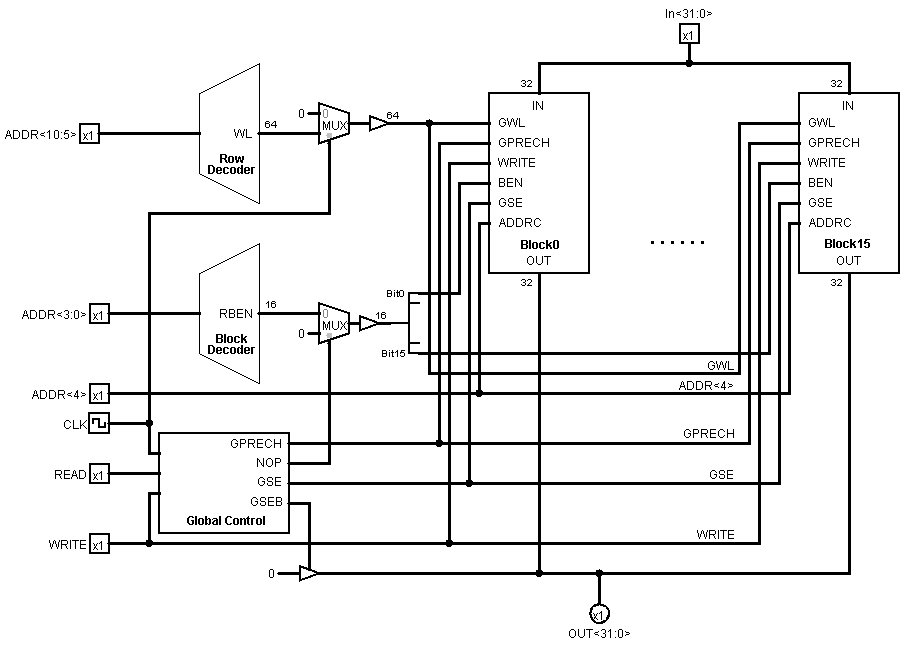
****

Figure 1 SRAM Architecture

Table 4 Annotation of SRAM-level signals and pins

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Signal/Pin** | **Width** | **Annotation** |
| I/O | IN<31:0> | 32 | Data Inputs |
| ADDR<10:5> | 6 | Row Address |
| ADDR<4> | 1 | Column Address |
| ADDR<3:0> | 4 | Block Address |
| READ | 1 | Read Control Signal |
| WRITE | 1 | Write Control Signal |
| OUT<31:0> | 32 | Data Outputs |
| Global Control | GPRECH | 1 | Global Pre-charge. Active high. |
| NOP | 1 | No Operation. Asserted when both READ and WRITE are low. |
| GSE | 1 | Global Sense-Amp Enable. |
| GSEB | 1 |  |
| Row Decoder | WL | 64 | Word Line - Row Decoder Output. Exactly one line is high. |
| Block Decoder | RBEN | 1 | (Raw) Block Enable – Block Decoder Output. |
| Block | GWL | 64 | Global Word Line. The asserted line will be high only during clock is low. |
|  | 1 | Block Enable. One bit per block. |
|  | 1 | Column Address. Feed by ADDR<4> |
|  | 32 | Block Output. Valid when and GSE are high, floating otherwise. |

1. **Global Control**

**Progress:**

* **Schematic**: Completed.
* **Layout:** Not done.
* **Simulation:** Various simulations have been done to assure functionality.

The global control block takes in clock and primary inputs and generates control signals for other parts of the SRAM. The detailed gate-level diagram for the Global Control block shown in Figure 1 is presented in Figure 2.

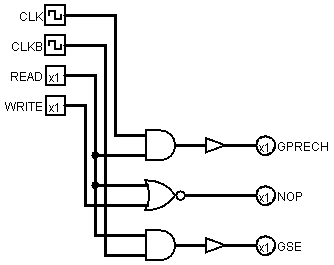


Figure 2 Gate-Level Schematic of the Global Control

1. **SRAM Block**

**Progress:**

* **Schematic**: Completed.
* **Layout:** Done for the SRAM array. Not done for the entire block structure.
* **Simulation:** Various simulations have been done to assure functionality.

Each SRAM block contains one bank (4kbit) of SRAM bit cell, and a set of peripherals that are necessary for the bit cell array to work. During each clock cycle, only one of the 16 blocks is on (controlled by BEN signal). When a block is not enabled, no pre-charge will ever occurs, and sense amplifiers are disabled. The output of a block if and only the block is enabled, and the local sense-amp enable is on (equivalently, the second half of a read cycle). The output is floating during any other circumstances.

Figure 3 recaps the transistor-level schematic of the SRAM bit cell we purposed in the proposal. Figure 4 shows the detail structure of a SRAM block, and Figure 5 shows the internal of the block control.

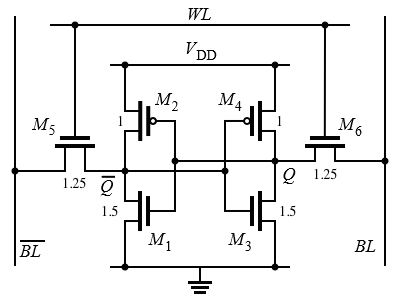


Figure 3 Transistor-Level Schematic of SRAM bit cell [1] (Adapted from Inductiveload)

As shown in Figure 4, instead of having 2 sense amplifiers, one for each word, we decided to insert the mux before the sense amplifier. Using this method, we are able to use one less 32-bit sense amplifier at the expense of one more 32-bit mux. Since we need only 2 transmission gates (4 transistors) per bit of a mux, while 8 transistors gates per bit of a sense amplifier, we saves considerable amount of area as we have 16 blocks. In addition, transmission gates does not consume energy directly while the sense amplifier do drain current from , we saves power at the same time.

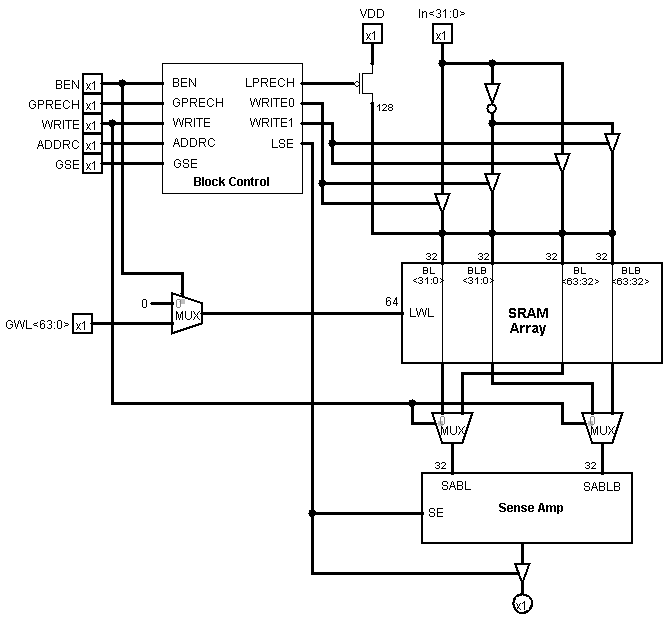


Figure 4 SRAM Block Structure

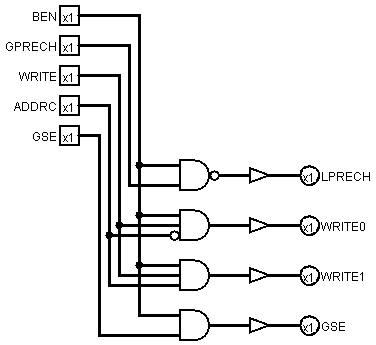


Figure 5 Gate-Level Schematic of the Block Control

1. **Decoder**

**Progress:**

* **Schematic**: Completed.
* **Layout:** In Progress.
* **Simulation:** Various simulations have been done to assure functionality.

There are two types of decoders in our project: 4to16 block address and 6to64 row address. Delay is our most sensitive metric for the entire circuit, but we have integrated the row decoder into the system such that as long as its delay is less than half a clock period, the delay for accessing the outputs is only 1 transmission gate (Figure 6). This drives all outputs to 0 at the beginning. Because of this implementation, the row decoder shouldn’t be part of the critical path, making delay a secondary factor. The block decoder is small enough to not be on the critical path too. This makes energy and area are priority metrics to decrease.

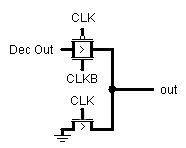


Figure 6 Transmission Schematic after Row Decoder

For the row decoder, the overall design involved a 2-stage hierarchy: pre decoder stage and the static CMOS AND the outputs. This drastically reduces the number of transistors. Table 5 shows the different architectures compared. Dynamic NOR and NAND prove to use too much energy and area, so static NOR will be used. Another test performed was adding a high voltage threshold (HVT) PMOS power gate to the inverters that made up the static CMOS AND gates. The small reduction in energy didn’t compensate for the larger increase in area. The same could be said when comparing the block decoder designs (Table 6). The final row decoder design is 2 3-to-8 static CMOS NOR predecoders (Figure 7). Further tests were conducted to make sure it was the best (Table 5).

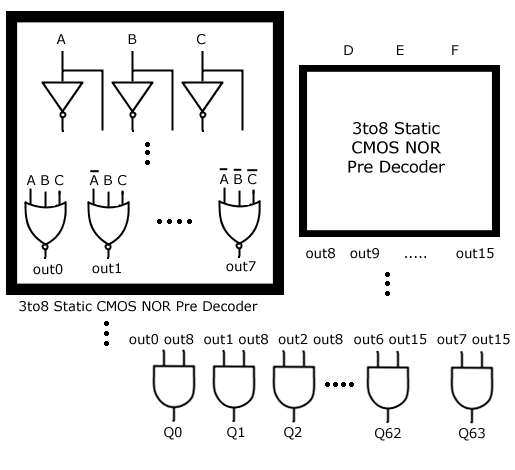
Table 5 Comparing Architectures for Row Decoder

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Delay ()** | **Energy ()** | **# of FET** |
| 2-to-4 DNOR Pre Dec. | N/A | 14.3 | 680 |
| 2-to-4 DNAND Pre Dec. | N/A | 2.47 | 656 |
| 2-to-4 NOR Pre Dec. | 102 | 0.715 | 572 |
| 2to4 NOR Pre Dec. with PMOS PG on Inverter | 114 | 0.681 | 636 |
| 3-to-8 NOR Pre. Dec. | 85.9 | 0.582 | 492 |

Table 6 Further Co

mparing Architectures for 3-to-8 NOR Pre Dec.

|  |  |  |  |
| --- | --- | --- | --- |
| **3-to-8 NOR Pre Dec** | **Delay ()** | **Energy ()** | **# of FET** |
| Original | 85.9 | 0.582 | 492 |
| PMOS PG on Inverter | 95.5 | 0.553 | 556 |
| PMOS PG on Inverter and HVT NMOS in NOR | 88.0 | 0.592 | 556 |
| Pass Gate (5 tran.) AND | 112 | 0.684 | 556 |

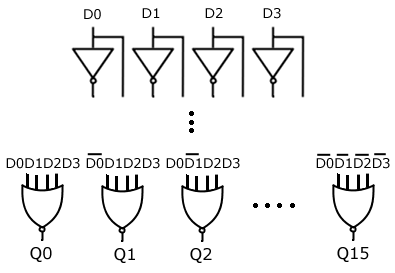


**Figure 7** High Level Schematic of Final Row Decoder

For the block decoder, Table 7 shows three architectures tested. The winner is a 4to16 decoder made out of static NORs (Figure 8). The current designs will have all minimal sized transistors for the lowest energy. The next steps to take are integrating the decoders into the system, and make them faster if they are on the critical path. The outputs of both decoders will be ANDed with each other to activate only one WL and conserve energy.

Table 7 Comparing Architectures for Block Decoder

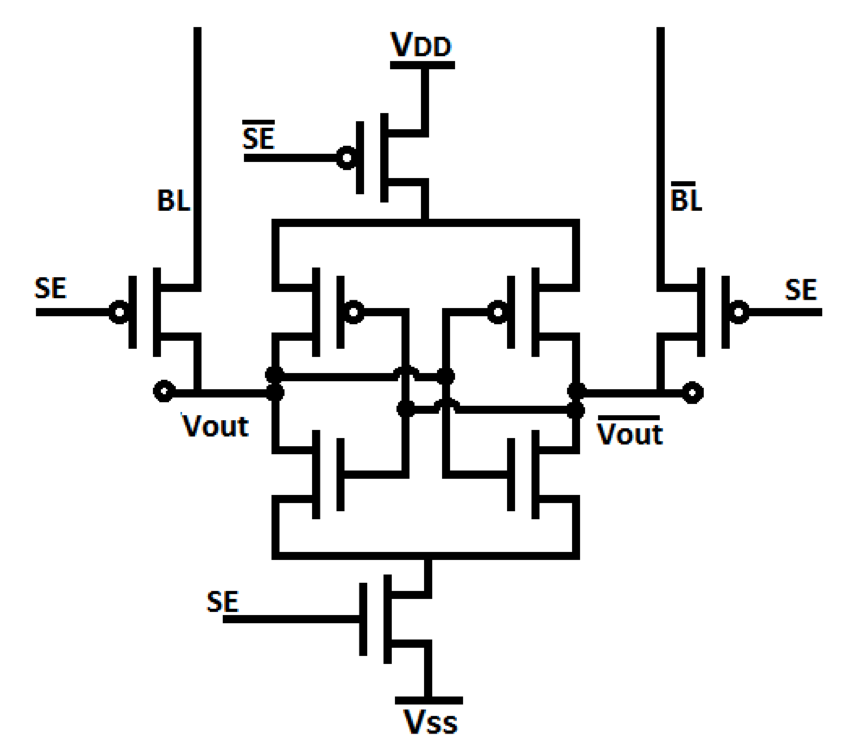
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Delay ()** | **Energy ()** | **# of FET** |
| 2-to-4 NOR Pre Dec. | 47.5 | 96.3 | 136 |
| 2-to-4 NOR Pre Dec. with PMOS PG on Inverter | 56.0 | 95.2 | 152 |
| 4to16 NOR | 41.1 | 84.3 | 136 |



**Figure 8** High Level Schematic of Final Block Decoder

1. **Sense Amplifier**

**Progress:**

* **Schematic**: Completed.
* **Layout:** Done for 1-bit sense amplifier.
* **Simulation:** Various simulations have been done to assure functionality.
* 
* Figure 7 Proposed Sense Amplifier Design

For the sense amplifier, we are using a latch-based voltage mode sense amplifier with added pass transistors to isolate the bitlines. As we have already justified our reasons for choosing this sense amplifier in the proposal, we won’t restate them here.

1. **SIMULATION MODEL**

To speed up the simulation, we built a block model to imitate a functional SRAM block. We decided that the block model needs to have the following properties.

* For the bits we care about, function should be ensured.
* Active and idle power of a modeled block should be close to those of a real block. That is, all inputs that are not always low should connect to the appropriate loads. All internal power consumption should be considered.
* Fast for simulation.

To accomplish these goals, we decided for the block model to use the same structure as shown in Figure 4, except that the SRAM array replaced by the array model shown in Figure 8. Using this schematic, we are able to preserve the correct function on row 0 and row 63, while dramatically reducing the circuit size using the option to represent the faked blocks.

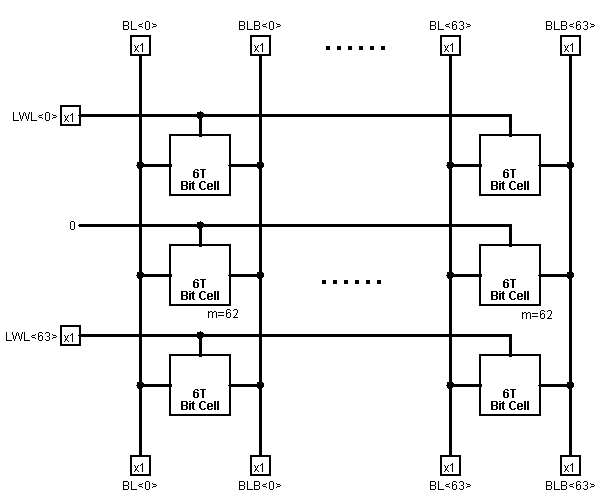


Figure 8 SRAM Array Model (To Replace the SRAM Array in Figure 4)

To further speed up the simulation, we use one modeled block to replace Block 15, and we use one modeled block with to replace all the other 15 blocks (Block0 – Block14), with hard wired to .

We simulated a READ cycle followed by a WRITE cycle and compare the energy consumption of one block between a real block and a modeled block. The result is shown in Table 8. Active and Idle indicates if the of the measured block is high or low.

Table 8 Comparison of Energy Consumption between a Full Block and a Modeled Block (Unit: )

|  |  |  |
| --- | --- | --- |
|  | **Full Block** | **Modeled Block** |
| **Active** | 2.7726 | 2.7717 |
| **Idle** | 3.5991 | 3.5983 |

With the results shown in Table 8, and simulation for the correct function on row 0 and row 63, we are confident that the modeled block correctly represents a real SRAM. It takes about 15 minutes to simulate one full block while it only take 1.5 minutes to simulate the entire SRAM with the block model, thus achieving a speed up of 160 times approximately.

1. **SIMULATION**
   1. **Static Noise Margin**

To examine the tolerance of the SRAM bit cell, simulations for static noise margins have been done. To evaluate hold SNM, word line is driven to 0 while BL and BLB are not driven. To evaluate read SNM, all the three terminals of a SRAM, WL, BL and BLB are driven high. During the simulation, a DC sweep was performed on node Q, and we export the values QB to a CSV file. We then wrote a python script to generate the butterfly plot and calculate the SNMs with the exported data.

Figure 9 and 10 shows the hold and the read butterfly curve, respectively. Both plots are for typical-typical case. Table 10 shows the SNMs for all the 5 process corners. In the table for the process corners, the first letter corresponds to PMOS, and the second letter corresponds to NMOS.

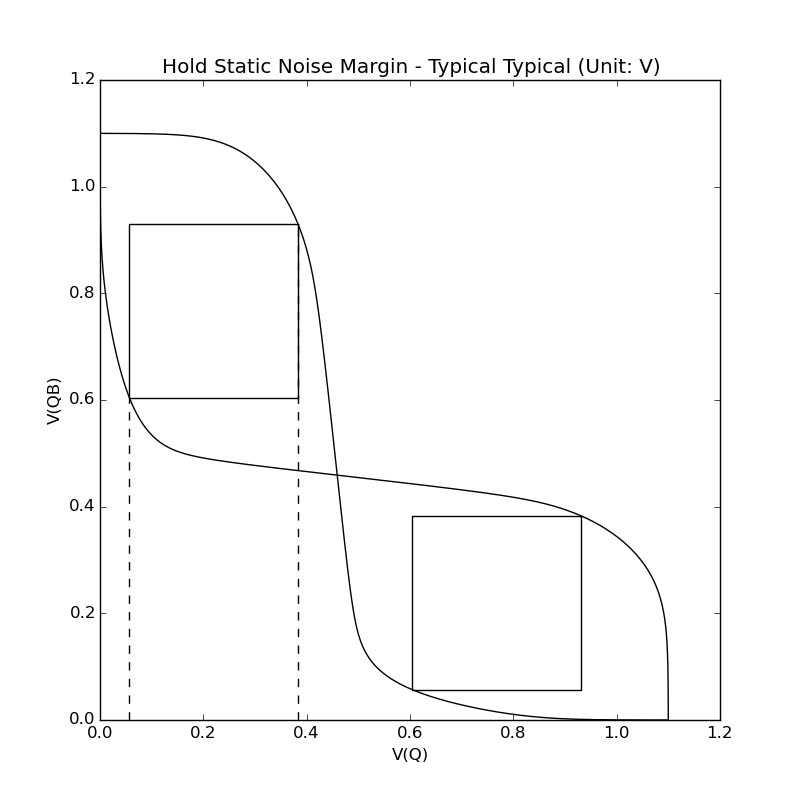


Figure 9 Hold Static Noise Margin Butterfly Plot (Typical-Typical Case)

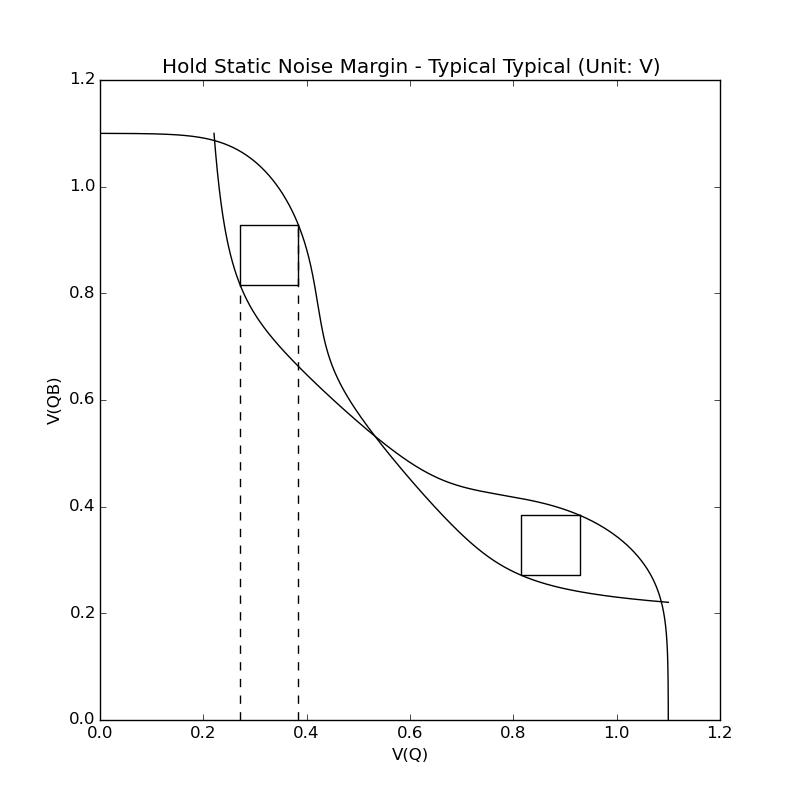


Figure 10 Read Static Noise Margin Butterfly Plot (Typical-Typical Case)

Table 9 Static Noise Margin at different global process corners – T=Typical; F=Fast; S=Slow; (Unit: mV)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | TT | FF | FS | SF | SS |
| HOLD SNM | 326 | 307 | 339 | 310 | 346 |
| READ SNM | 113 | 92 | 140 | 84 | 134 |

* 1. **Simulation Results**

Simulation results are included in the appendix.

1. **REFERENCE**

[1] Inductiveload (2009, August). *SRAM Cell (6 Transistors)*. Retrieved from http://upload.wikimedia.org/wikipedia/commons/3/31/SRAM\_Cell\_%286\_Transistors%29.svgt